



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **09288580 A**(43) Date of publication of application: **04.11.97**

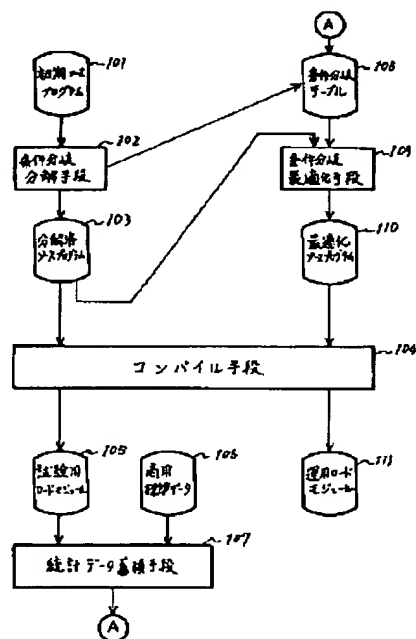
(51) Int. Cl.

G06F 9/45(21) Application number: **08098045**(71) Applicant: **NEC CORP**(22) Date of filing: **19.04.96**(72) Inventor: **TANAKA SEIICHI****(54) DEVICE AND METHOD FOR OPTIMIZING SOURCE PROGRAM****(57) Abstract:**

PROBLEM TO BE SOLVED: To prepare an executable load module at high speed by optimizing a source program containing a complicated multi-stage instruction.

SOLUTION: A condition branch decomposing means 102 analyzes the condition branching instruction of the source program and prepares a condition branch table 108 containing the information on correspondence between processing for each condition in the condition branching instruction of the source program and the number of executed processes, and a statistic data storage means 107 executes a load module 105 for test, which is prepared by compiling the source program, while using data for test so as to measures the number of respective executed processes and to store the number of executed processes in the condition branch table 108. Based on the number of respective executed processes in the condition branch table 108, a condition branch optimizing means 109 rearranges the processes for each condition in the condition branching instruction so that the source program can be optimized.

COPYRIGHT: (C)1997,JPO



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平9-288580

(43)公開日 平成9年(1997)11月4日

(51)Int.Cl.⁶
G 0 6 F 9/45

識別記号 庁内整理番号

F I
G 0 6 F 9/44

技術表示箇所

3 2 2 F

審査請求 有 請求項の数14 O L (全 13 頁)

(21)出願番号 特願平8-98045

(22)出願日 平成8年(1996)4月19日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 田中 精一

東京都港区芝五丁目7番1号 日本電気株式会社内

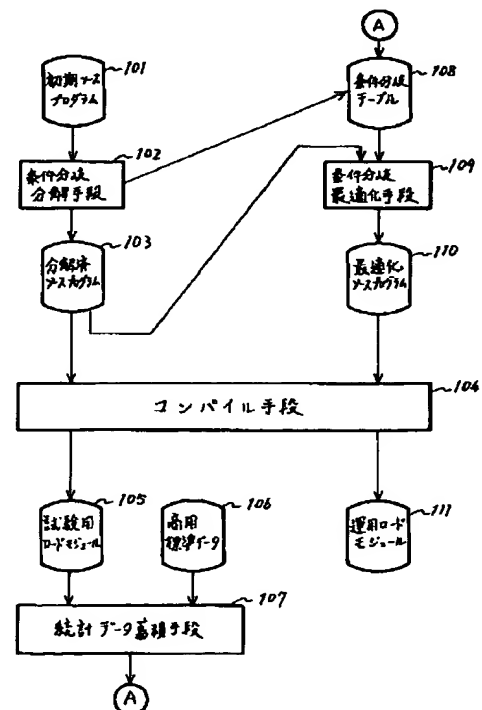
(74)代理人 弁理士 京本 直樹 (外2名)

(54)【発明の名称】 ソースプログラムの最適化装置および最適化方法

(57)【要約】

【課題】複雑な多段階命令を含むソースプログラムを最適化して高速に実行可能なロードモジュールを作成する。

【解決手段】条件分岐分解手段が、ソースプログラムの条件分岐命令を解析するとともに、ソースプログラムの条件分岐命令内の各条件毎の処理とその実行件数との対応情報を含む条件分岐テーブルを作成し、統計データ蓄積手段が、ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測して実行件数を条件分岐テーブルに記憶し、条件分岐最適化手段が、条件分岐テーブル内の各処理の実行件数に基づいて、ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する。



【特許請求の範囲】

【請求項1】 ソースプログラムの条件分岐命令内の各条件毎の処理を実行頻度の順に並べ替えることにより、ソースプログラムを最適化する条件分岐最適化手段を備えることを特徴とするソースプログラムの最適化装置。

【請求項2】 ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理とその実行件数との対応情報を含む条件分岐テーブルを作成する条件分岐分解手段と、前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測し、該実行件数を前記条件分岐テーブルに記憶する統計データ蓄積手段と、前記条件分岐テーブル内の各処理の実行件数に基づいて、前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する条件分岐最適化手段とを備えることを特徴とするソースプログラムの最適化装置。

【請求項3】 ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理とその処理を行う条件とその実行件数との対応情報を含む条件分岐テーブルを作成する条件分岐分解手段と、前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測し、該実行件数を前記条件分岐テーブルに記憶する統計データ蓄積手段と、前記条件分岐テーブル内の各処理を行う条件に基づいて前記ソースプログラムにおける条件分岐命令を多肢選択型の条件分岐命令に変換するとともに、前記条件分岐テーブル内の各処理の実行件数に基づいて前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する条件分岐最適化手段とを備えることを特徴とするソースプログラムの最適化装置。

【請求項4】 前記条件分岐分解手段は、初期ソースプログラムの二者択一型の条件分岐文の直下に選択肢の一方が存在しない場合に、その位置にダミー処理を挿入するとともにチェックポイントを設定して分解済ソースプログラムを作成した後、該ダミー処理を前記条件分岐テーブルに登録し、前記統計データ蓄積手段は、前記分解済ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行する際に、該試験用データが前記チェックポイントを通過する件数により各処理の実行件数を計測し、該実行件数を前記条件分岐テーブルに記憶することを特徴とする請求項2記載のソースプログラムの最適化装置。

【請求項5】 前記条件分岐分解手段は、さらに、初期ソースプログラムの二者択一型の条件分岐文の直下に選

択肢の一方が存在する場合に、自身の処理を再帰的に呼び出すことを特徴とする請求項4記載のソースプログラムの最適化装置。

【請求項6】 前記条件分岐分解手段は、さらに、選択肢の直下に二者択一型の条件分岐文が存在しない場合に、その位置にチェックポイントを設定して分解済ソースプログラムを作成した後、その位置にある処理を前記条件分岐テーブルに登録することを特徴とする請求項5記載のソースプログラムの最適化装置。

10 【請求項7】 前記条件分岐最適化手段は、初期ソースプログラムの二者択一型の条件分岐文を多肢選択型の条件分岐文に変換した後、前記条件分岐テーブルにおける実行件数の多い順に対応する条件の記述を行い、さらに、その条件が満たされる場合に行うべき処理を記述することにより、最適化ソースプログラムを作成することを特徴とする請求項6記載のソースプログラムの最適化装置。

【請求項8】 ソースプログラムの条件分岐命令内の各条件毎の処理を実行頻度の順に並べ替えることにより、ソースプログラムを最適化するステップを含むことを特徴とするソースプログラムの最適化方法。

20 【請求項9】 ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理とその実行件数との対応情報を記憶する第1のステップと、前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測して記憶する第2のステップと、

30 前記第2のステップで記憶した実行件数に基づいて、前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する第3のステップとを含むことを特徴とするソースプログラムの最適化方法。

【請求項10】 ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理とその処理を行う条件とその実行件数との対応情報を記憶する第1のステップと、

40 前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測して記憶する第2のステップと、

前記第1のステップで記憶した条件に基づいて前記ソースプログラムにおける条件分岐命令を多肢選択型の条件分岐命令に変換するとともに、前記第2のステップで記憶した実行件数に基づいて前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する第3のステップとを含むことを特徴とするソースプログラムの最適化方法。

【請求項 1 1】 前記第 1 のステップで、初期ソースプログラムの二者択一型の条件分岐文の直下に選択肢の一方が存在しない場合に、その位置にダミー処理を挿入するとともにチェックポイントを設定して分解済ソースプログラムを作成した後、該ダミー処理を記憶し、前記第 2 のステップで、前記分解済ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行する際に、該試験用データが前記チェックポイントを通過する件数により各処理の実行件数を計測して記憶することを特徴とする請求項 1 0 記載のソースプログラムの最適化方法。

【請求項 1 2】 前記第 1 のステップで、さらに、初期ソースプログラムの二者択一型の条件分岐文の直下に選択肢の一方が存在する場合に、自身の処理を再帰的に呼び出すことを特徴とする請求項 1 1 記載のソースプログラムの最適化方法。

【請求項 1 3】 前記第 1 のステップで、さらに、選択肢の直下に二者択一型の条件分岐文が存在しない場合に、その位置にチェックポイントを設定して分解済ソースプログラムを作成した後、その位置にある処理を記憶

【請求項 1 4】 前記第 3 のステップで、初期ソースプログラムの二者択一型の条件分岐文を多肢選択型の条件分岐文に変換した後、前記条件分岐テーブルにおける実行件数の多い順に対応する条件の記述を行い、さらに、その条件が満たされる場合に行うべき処理を記述することにより、最適化ソースプログラムを作成することを特徴とする請求項 1 3 記載のソースプログラムの最適化方法。

【発明の詳細な説明】

【発明の属する技術分野】本発明は、ソースプログラムを最適化し、高速に実行可能なロードモジュールを生成するソースプログラムの最適化装置および最適化方法に関する。

【0001】

【従来の技術】条件分岐命令を含むプログラムを高速に実行するため、そのプログラムの実行時に、条件分岐命令を構成する複数の条件節をその成立頻度の高い順に並べ替える方法が、特開平 2-2425 号公報に記載されている。この方法では、BASIC 言語において、複数の IF 文からなる多方向分岐命令の高速化について、記載されている。

【0002】

【発明が解決しようとする課題】しかし、この従来の技術においては、IF 文からなる多方向分岐命令については検討されているが、複数の IF 文がいわゆる入れ子の構造をなす多段階分岐命令については考慮されていない。したがって、複雑な多段階分岐命令に対しては、高速化できないという問題点があった。

【0003】また、BASIC 言語を前提としていることからプログラムの実行時に条件節の並べ替えを行っており、高速化されたプログラムの内容をソースレベルでユーザに提供することができないという問題点もあった。

【0004】本発明の目的は、複雑な多段階命令を含むソースプログラムを最適化して高速に実行可能なロードモジュールを作成することにある。

【0005】本発明の他の目的は、最適化されたソースプログラムの記述内容をユーザに提供できるようにすることにある。

【0006】

【課題を解決するための手段】本発明の第 1 のソースプログラムの最適化装置は、ソースプログラムの条件分岐命令内の各条件毎の処理を実行頻度の順に並べ替えることにより、ソースプログラムを最適化する条件分岐最適化手段から構成されている。

【0007】本発明の第 2 のソースプログラムの最適化装置は、ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理とその実行件数との対応情報を含む条件分岐テーブルを作成する条件分岐分解手段と、前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測し、該実行件数を前記条件分岐テーブルに記憶する統計データ蓄積手段と、前記条件分岐テーブル内の各処理の実行件数に基づいて、前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する条件分岐最適化手段とから構成されている。

【0008】本発明の第 3 のソースプログラムの最適化装置は、ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理とその処理を行う条件とその実行件数との対応情報を含む条件分岐テーブルを作成する条件分岐分解手段と、前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測し、該実行件数を前記条件分岐テーブルに記憶する統計データ蓄積手段と、前記条件分岐テーブル内の各処理を行う条件に基づいて前記ソースプログラムにおける条件分岐命令を多肢選択型の条件分岐命令に変換するとともに、前記条件分岐テーブル内の各処理の実行件数に基づいて前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する条件分岐最適化手段とから構成されている。

【0009】本発明の第 4 のソースプログラムの最適化装置は、第 2 のソースプログラムの最適化装置において、前記条件分岐分解手段は、初期ソースプログラムの二者択一型の条件分岐文の直下に選択肢の一方が存在し

ない場合に、その位置にダミー処理を挿入するとともにチェックポイントを設定して分解済ソースプログラムを作成した後、該ダミー処理を前記条件分岐テーブルに登録し、前記統計データ蓄積手段は、前記分解済ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行する際に、該試験用データが前記チェックポイントを通過する件数により各処理の実行件数を計測し、該実行件数を前記条件分岐テーブルに記憶することを特徴とする。

【0010】本発明の第5のソースプログラムの最適化装置は、第4のソースプログラムの最適化装置において、前記条件分岐分解手段は、さらに、初期ソースプログラムの二者択一型の条件分岐文の直下に選択肢の一方が存在する場合に、自身の処理を再帰的に呼び出すことを特徴とする。

【0011】本発明の第6のソースプログラムの最適化装置は、第5のソースプログラムの最適化装置において、前記条件分岐分解手段は、さらに、選択肢の直下に二者択一型の条件分岐文が存在しない場合に、その位置にチェックポイントを設定して分解済ソースプログラムを作成した後、その位置にある処理を前記条件分岐テーブルに登録することを特徴とする。

【0012】本発明の第7のソースプログラムの最適化装置は、第6のソースプログラムの最適化装置において、前記条件分岐最適化手段は、初期ソースプログラムの二者択一型の条件分岐文を多肢選択型の条件分岐文に変換した後、前記条件分岐テーブルにおける実行件数の多い順に対応する条件の記述を行い、さらに、その条件が満たされる場合に行うべき処理を記述することにより、最適化ソースプログラムを作成することを特徴とする。

【0013】本発明の第1のソースプログラムの最適化方法は、ソースプログラムの条件分岐命令内の各条件毎の処理を実行頻度の順に並べ替えることにより、ソースプログラムを最適化するステップを含んでいる。

【0014】本発明の第2のソースプログラムの最適化方法は、ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理とその実行件数との対応情報を記憶する第1のステップと、前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測して記憶する第2のステップと、前記第2のステップで記憶した実行件数に基づいて、前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する第3のステップとを含んでいる。

【0015】本発明の第3のソースプログラムの最適化方法は、ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理と

その処理を行う条件とその実行件数との対応情報を記憶する第1のステップと、前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測して記憶する第2のステップと、前記第1のステップで記憶した条件に基づいて前記ソースプログラムにおける条件分岐命令を多肢選択型の条件分岐命令に変換するとともに、前記第2のステップで記憶した実行件数に基づいて前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する第3のステップとを含んでいる。

【0016】本発明の第4のソースプログラムの最適化方法は、第3のソースプログラムの最適化方法において、前記第1のステップで、初期ソースプログラムの二者択一型の条件分岐文の直下に選択肢の一方が存在しない場合に、その位置にダミー処理を挿入するとともにチェックポイントを設定して分解済ソースプログラムを作成した後、該ダミー処理を記憶し、前記第2のステップで、前記分解済ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行する際に、該試験用データが前記チェックポイントを通過する件数により各処理の実行件数を計測して記憶することを特徴とする。

【0017】本発明の第5のソースプログラムの最適化方法は、第4のソースプログラムの最適化方法において、前記第1のステップで、さらに、初期ソースプログラムの二者択一型の条件分岐文の直下に選択肢の一方が存在する場合に、自身の処理を再帰的に呼び出すことを特徴とする。

【0018】本発明の第6のソースプログラムの最適化方法は、第5のソースプログラムの最適化方法において、前記第1のステップで、さらに、選択肢の直下に二者択一型の条件分岐文が存在しない場合に、その位置にチェックポイントを設定して分解済ソースプログラムを作成した後、その位置にある処理を記憶することを特徴とする。

【0019】本発明の第7のソースプログラムの最適化方法は、第6のソースプログラムの最適化方法において、前記第3のステップで、初期ソースプログラムの二者択一型の条件分岐文を多肢選択型の条件分岐文に変換した後、前記条件分岐テーブルにおける実行件数の多い順に対応する条件の記述を行い、さらに、その条件が満たされる場合に行うべき処理を記述することにより、最適化ソースプログラムを作成することを特徴とする。

【0020】本発明の第1のコンピュータにおいて読み取り可能な媒体は、ソースプログラムの条件分岐命令内の各条件毎の処理を実行頻度の順に並べ替えることにより、ソースプログラムを最適化するステップを含むプログラムを格納する。

【0021】本発明の第2のコンピュータにおいて読み

取り可能な媒体は、ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理とその実行件数との対応情報を記憶する第1のステップと、前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測して記憶する第2のステップと、前記第2のステップで記憶した実行件数に基づいて、前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する第3のステップとを含むプログラムを格納する。

【0022】本発明の第3のコンピュータにおいて読み取り可能な媒体は、ソースプログラムの条件分岐命令を解析し、該ソースプログラムの条件分岐命令内の各条件毎の処理とその処理を行う条件とその実行件数との対応情報を記憶する第1のステップと、前記ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行することにより各処理の実行件数を計測して記憶する第2のステップと、前記第1のステップで記憶した条件に基づいて前記ソースプログラムにおける条件分岐命令を多肢選択型の条件分岐命令に変換するとともに、前記第2のステップで記憶した実行件数に基づいて前記ソースプログラムにおける条件分岐命令内の各条件毎の処理を並べ替えることにより、ソースプログラムを最適化する第3のステップとを含むプログラムを格納する。

【0023】本発明の第4のコンピュータにおいて読み取り可能な媒体は、第3のコンピュータにおいて読み取り可能な媒体に格納されたプログラムにおいて、前記第1のステップで、初期ソースプログラムの二者択一型の条件分岐文の直下に選択肢の一方が存在しない場合に、その位置にダミー処理を挿入するとともにチェックポイントを設定して分解済ソースプログラムを作成した後、該ダミー処理を記憶し、前記第2のステップで、前記分解済ソースプログラムをコンパイルして作成された試験用ロードモジュールを試験用データを用いて実行する際に、該試験用データが前記チェックポイントを通過する件数により各処理の実行件数を計測して記憶することを特徴とする。

【0024】本発明の第5のコンピュータにおいて読み取り可能な媒体は、第4のコンピュータにおいて読み取り可能な媒体に格納されたプログラムにおいて、前記第1のステップで、さらに、初期ソースプログラムの二者択一型の条件分岐文の直下に選択肢の一方が存在する場合に、自身の処理を再帰的に呼び出すことを特徴とする。

【0025】本発明の第6のコンピュータにおいて読み取り可能な媒体は、第5のコンピュータにおいて読み取り可能な媒体に格納されたプログラムにおいて、前記第1のステップで、さらに、選択肢の直下に二者択一型の

条件分岐文が存在しない場合に、その位置にチェックポイントを設定して分解済ソースプログラムを作成した後、その位置にある処理を記憶することを特徴とする。

【0026】本発明の第7のコンピュータにおいて読み取り可能な媒体は、第6のコンピュータにおいて読み取り可能な媒体に格納されたプログラムにおいて、前記第3のステップで、初期ソースプログラムの二者択一型の条件分岐文を多肢選択型の条件分岐文に変換した後、前記条件分岐テーブルにおける実行件数の多い順に対応する条件の記述を行い、さらに、その条件が満たされる場合に行うべき処理を記述することにより、最適化ソースプログラムを作成することを特徴とする。

【0027】

【発明の実施の形態】次に、本発明の一実施例について図面を参照して詳細に説明する。

【0028】図1を参照すると、本発明の一実施例であるソースプログラムの最適化装置、あるいは最適化方法を適用したシステムは、初期ソースプログラム101の条件分岐命令を解析して分解済ソースプログラム103を作成するとともに、条件分岐命令内の各条件毎の処理とその実行件数との対応情報を含む条件分岐テーブル108を作成する条件分岐分解手段102と、分解済ソースプログラム103をコンパイルして作成された試験用ロードモジュール105を商用標準データ106を用いて実行することにより各処理の実行件数を計測し、実行件数を条件分岐テーブル108に記憶する統計データ蓄積手段107と、条件分岐テーブル108内の各処理の実行件数に基づいて、分解済ソースプログラム103における条件分岐命令内の各条件毎の処理を並べ替えることにより、最適化ソースプログラム110を作成する条件分岐最適化手段109とを備えている。

【0029】次に、本発明の一実施例であるソースプログラムの最適化装置、あるいは最適化方法を適用したシステムの動作について、図1～図7により説明する。

【0030】図1を参照すると、条件分岐分解手段102は、初期ソースプログラム101を分解し、分解済ソースプログラム103を作成するとともに、条件分岐テーブル108に条件分岐データを登録する。条件分岐分解手段102の動作について、図2～図3を参照して説明する。

【0031】図2に、初期ソースプログラム101、分解済ソースプログラム103、条件分岐テーブル108の一例を示す。

【0032】この例では、初期ソースプログラム101は、条件A、B、Cを有しており、条件Aが真かつ条件Bが真の時に処理Pを、条件Aが真かつ条件Bが偽かつ条件Cが偽の時に処理Qを実行するように組まれている。

【0033】図3は、条件分岐分解手段102が初期ソースプログラム101を分解して分解済ソースプログラ

ム103を作成する際の動作をフローチャートにより示したものである。

【0034】条件分岐分解手段102は、最適化の対象である初期ソースプログラム101を読み込み、IF文を検索することにより条件分岐命令であるか否かを判断する（ステップ301）。

【0035】条件分岐命令である場合は、その下にTHEN文が存在するか否かを判断する（ステップ302）。THEN文が存在する場合は、ステップ301～ステップ311の処理を再帰的に呼び出す（ステップ303）。一方、THEN文が存在しない場合は、ソースプログラム上のその位置にTHEN文及びダミー処理を挿入し（ステップ304）、チェックポイントを設定する（ステップ305）とともに、条件及び処理の情報を条件分岐テーブル108に登録する（ステップ306）。

【0036】次に、現在解析を行っている条件分岐命令（必ずしも最上位の条件分岐命令とは限らず、再帰的に呼び出されている場合は、下位の条件分岐命令である可能性もある。）の下にELSE文が存在するか否かを判断する（ステップ307）。ELSE文が存在する場合は、ステップ301～ステップ311の処理を再帰的に呼び出す（ステップ308）。一方、ELSE文が存在しない場合は、ソースプログラム上のその位置にELSE文及びダミー処理を挿入し（ステップ309）、チェックポイントを設定する（ステップ310）とともに、条件及び処理の情報を条件分岐テーブル108に登録する（ステップ311）。

【0037】また、ステップ301で条件分岐命令でないと判断された場合には、読み込んだ処理の位置にチェックポイントを設定する（ステップ312）とともに、条件及び処理の情報を条件分岐テーブル108に登録する（ステップ313）。

【0038】このようにして、分解済ソースプログラム103が作成される。

【0039】図2を参照すると、分解済ソースプログラム103においては、条件Aが真かつ条件Bが偽かつ条件Cが真の位置にダミー処理Xが挿入され、条件Aが偽の位置にダミー処理Yが挿入されている。さらに、処理Pに対してチェックポイント1が、ダミー処理Xに対してチェックポイント2が、処理Qに対してチェックポイント3が、ダミー処理Yに対してチェックポイント4が設定されている。

【0040】ここで、ダミー処理とは、処理内容を伴わない処理であり、実際には何も実行されず、条件をBREAKするだけのものである。

【0041】また、図2を参照すると、条件分岐テーブル108は、初期ソースプログラム101に含まれる条件A、B、Cの真偽と、その時に行う処理との対応情報を格納する。条件分岐テーブル中の数字「1」は、対応

する条件が真の時に対応する処理を実行することを示し、数字「0」は、対応する条件が偽の時に対応する処理を実行することを示す。また、「NULL」は、対応する条件を判断せずに対応する処理を実行することを示している。したがって、例えば、条件Aが真かつ条件Bが真の時に処理Pを実行することが分かる。

【0042】なお、本実施例の条件分岐テーブルでは、条件がA、B、Cの3つであるが、条件の数は必要に応じて設定できる。

10 【0043】また、図1を参照すると、コンパイル手段104は、分解済ソースプログラム103をコンパイルして、試験用ロードモジュール105を作成する。そして、統計データ蓄積手段107は、商用標準データ106を使用して試験用ロードモジュール105を実行することにより、条件分岐テーブル108を更新する。統計データ蓄積手段107の動作について、図4～図5を参照して説明する。

20 【0044】図4に、試験用ロードモジュール105、商用標準データ106、条件分岐テーブル108の一例を示す。なお、試験用ロードモジュール105については、便宜上、ソースプログラムのイメージで示している。

【0045】この例では、試験用ロードモジュール105は、条件Aが真かつ条件Bが真の時に処理Pを実行し、条件Aが真かつ条件Bが偽かつ条件Cが偽の時に処理Qを実行し、条件Aが真かつ条件Bが偽かつ条件Cが真の時にダミー処理Xを実行し、条件Aが偽の時にダミー処理Yを実行するように組まれている。

30 【0046】また、商用標準データ106は、各条件に対して図に示すような分岐を行うデータから構成されている。例えば、1件目のデータは、条件Aが真かつ条件Bが偽かつ条件Cが真となるようなデータである。また、2件目のデータは、条件Aが偽となるようなデータである。

【0047】図5は、統計データ蓄積手段107が試験用ロードモジュール105と商用標準データ106に基づいて条件分岐テーブル108を更新する際の動作をフローチャートにより示したものである。

40 【0048】まず、統計データ蓄積手段107は、条件分岐テーブル108における件数の初期化を行う（ステップ501）。

【0049】そして、商用標準データ106を1件読み込み（ステップ502）、条件分岐テーブル108において、この時通過したチェックポイントに対応する件数に1を加算する（ステップ503）。

【0050】さらに、統計データ蓄積手段107は、未処理のデータが他にあるか否かを判断し（ステップ504）、ある場合には、ステップ502～ステップ503の処理を繰り返す。

50 【0051】このようにして、条件分岐テーブル108

の更新が行われる。

【0052】図4を参照すると、条件分岐テーブル108は、処理がチェックポイント1を通過した件数が2件、チェックポイント2を通過した件数が8件、チェックポイント3を通過した件数が10件、チェックポイント4を通過した件数が5件であることを示している。すなわち、処理Pによって処理される件数が2件、ダミー処理Xによって処理される件数が8件、処理Qによって処理される件数が10件、ダミー処理Yによって処理される件数が5件であることを示している。

【0053】さらに、図1を参照すると、条件分岐最適化手段109は、分解済ソースプログラム103を読み込み、条件分岐テーブル108に基づいて最適化ソースプログラム110を作成する。条件分岐最適化手段109の動作について、図6～図7を参照して説明する。

【0054】図6に、分解済ソースプログラム103、条件分岐テーブル108、最適化ソースプログラム110の一例を示す。

【0055】この例では、分解済ソースプログラム103においては、条件Aが真かつ条件Bが偽かつ条件Cが真の位置にダミー処理Xが挿入され、条件Aが偽の位置にダミー処理Yが挿入されている。さらに、処理Pに対してチェックポイント1が、ダミー処理Xに対してチェックポイント2が、処理Qに対してチェックポイント3が、ダミー処理Yに対してチェックポイント4が設定されている。

【0056】また、条件分岐テーブル108は、処理がチェックポイント1を通過した件数が2件、チェックポイント2を通過した件数が8件、チェックポイント3を通過した件数が10件、チェックポイント4を通過した件数が5件であることを示している。すなわち、処理Pによって処理される件数が2件、ダミー処理Xによって処理される件数が8件、処理Qによって処理される件数が10件、ダミー処理Yによって処理される件数が5件であることを示している。

【0057】図7は、条件分岐最適化手段109が分解済ソースプログラム103と条件分岐テーブルに基づいて最適化ソースプログラム110を作成する際の動作をフローチャートにより示したものである。

【0058】条件分岐最適化手段109は、条件分岐テーブル108を件数の大きい順に並べ替える（ステップ701）。

【0059】図6に示すような条件分岐テーブル108に対しては、チェックポイント3を通過する処理Qの件数が1番多いので、処理Qが1番目になるようにし、次いで、ダミー処理X、ダミー処理Y、処理Pの順になるように並べ替える。

【0060】また、条件分岐最適化手段109は、IF～THEN～ELSEによって記述されたソースプログラムの形式をCASE～OFに変換する（ステップ70

2）。

【0061】次に、条件分岐テーブル108から、1つの処理に相当するレコードを読み込む（ステップ703）。さらに、その処理を実行すべき条件を記述した後（ステップ704）、その処理を記述する（ステップ705）。

【0062】さらに、条件分岐最適化手段109は、未処理の条件節が他にあるか否かを判断し（ステップ706）、ある場合には、ステップ701～ステップ705の処理を繰り返す。

【0063】このようにして、最適化ソースプログラム110が作成される。

【0064】図6を参照すると、最適化ソースプログラム110は、条件Aが真かつ条件Bが偽かつ条件Cが偽の時に処理Qを実行し、条件Aが真かつ条件Bが偽かつ条件Cが真の時にダミー処理Xを実行し、条件Aが偽の時にダミー処理Yを実行し、条件Aが真かつ条件Bが真の時に処理Pを実行するように組まれている。

【0065】図1を参照すると、最後に、コンパイル手段104は、最適化ソースプログラム110をコンパイルして、運用ロードモジュール111を作成する。

【0066】以上により、本発明の一実施例であるソースプログラムの最適化装置、あるいは最適化方法を適用したシステムの処理が終了する。

【0067】なお、本実施例においては、条件分岐命令として、IF文、THEN文、ELSE文を含むものを考えたが、必ずしもこれらの文である必要はなく、本発明は、IF文、THEN文、ELSE文の機能に相当する他の文についても適用することができる。また、ソースプログラム中のIF文等の二者択一型の条件分岐文だけでなく、CASE文等の多肢選択型の条件分岐文に対しても、同様にして、本発明を適用することが可能である。

【0068】本発明の一実施例であるソースプログラムの最適化装置、あるいは最適化方法を適用したシステムは、複雑な条件分岐を多岐条件分岐に変換するとともに、条件節をその成立頻度の順に並べ変えることにより、プログラムを高速に実行できるようにすることができるという効果を有している。

【0069】また、複雑な条件分岐を多岐条件分岐に変換し、条件節をその成立頻度の順に並べ変えた後にコンパイルを行うことにより、高速に実行可能に変更されたソースプログラムを提供することができるという効果を有している。

【0070】

【発明の効果】以上説明したように、本発明のソースプログラムの最適化装置および最適化方法は、プログラムを高速に実行できるようにすることができるという効果を有している。

【0071】また、高速に実行可能に変更されたソース

プログラムを提供することができるという効果を有している。

【図面の簡単な説明】

【図1】図1は本発明の一実施例の構成を示すブロック図である。

【図2】図2は本発明の一実施例における条件分岐分解手段102の動作の概要を示す図である。

【図3】図3は本発明の一実施例における条件分岐分解手段102の処理を示す流れ図である。

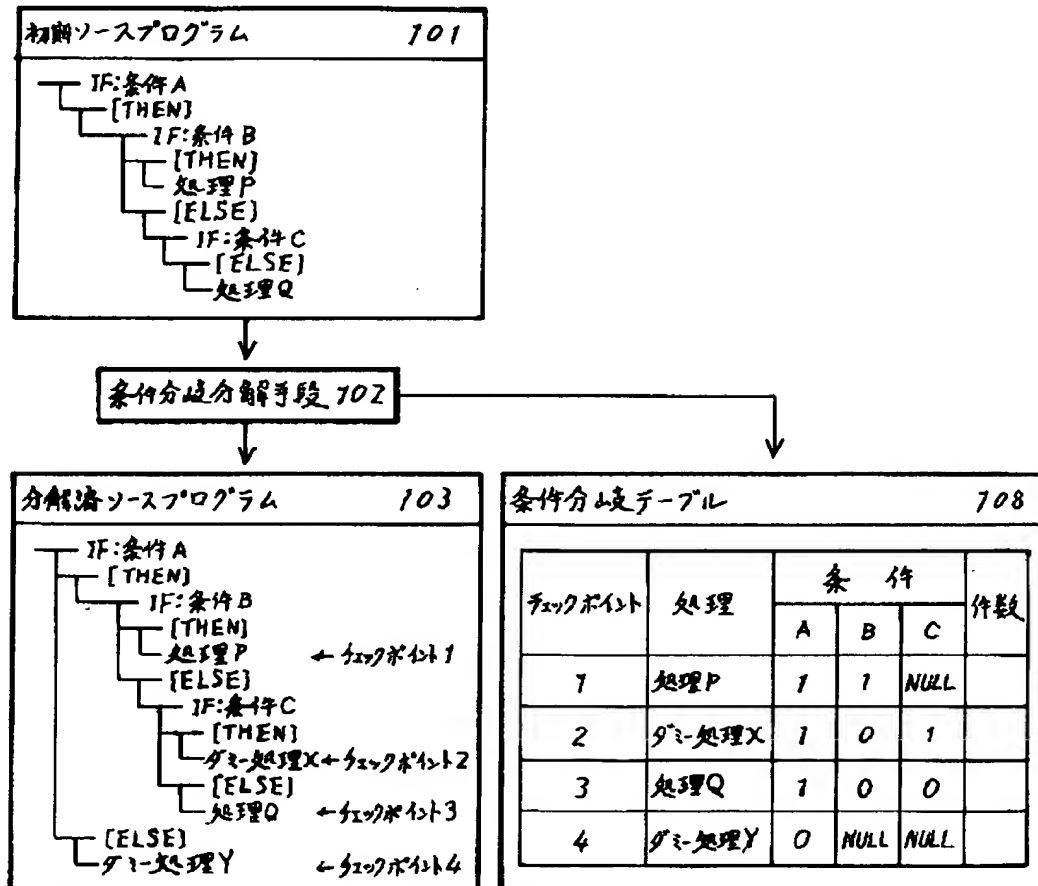
【図4】図4は本発明の一実施例における統計データ蓄積手段107の動作の概要を示す図である。

【図5】図5は本発明の一実施例における統計データ蓄積手段107の処理を示す流れ図である。

【図6】図6は本発明の一実施例における条件分岐最適化手段109の動作の概要を示す図である。

*

【図2】



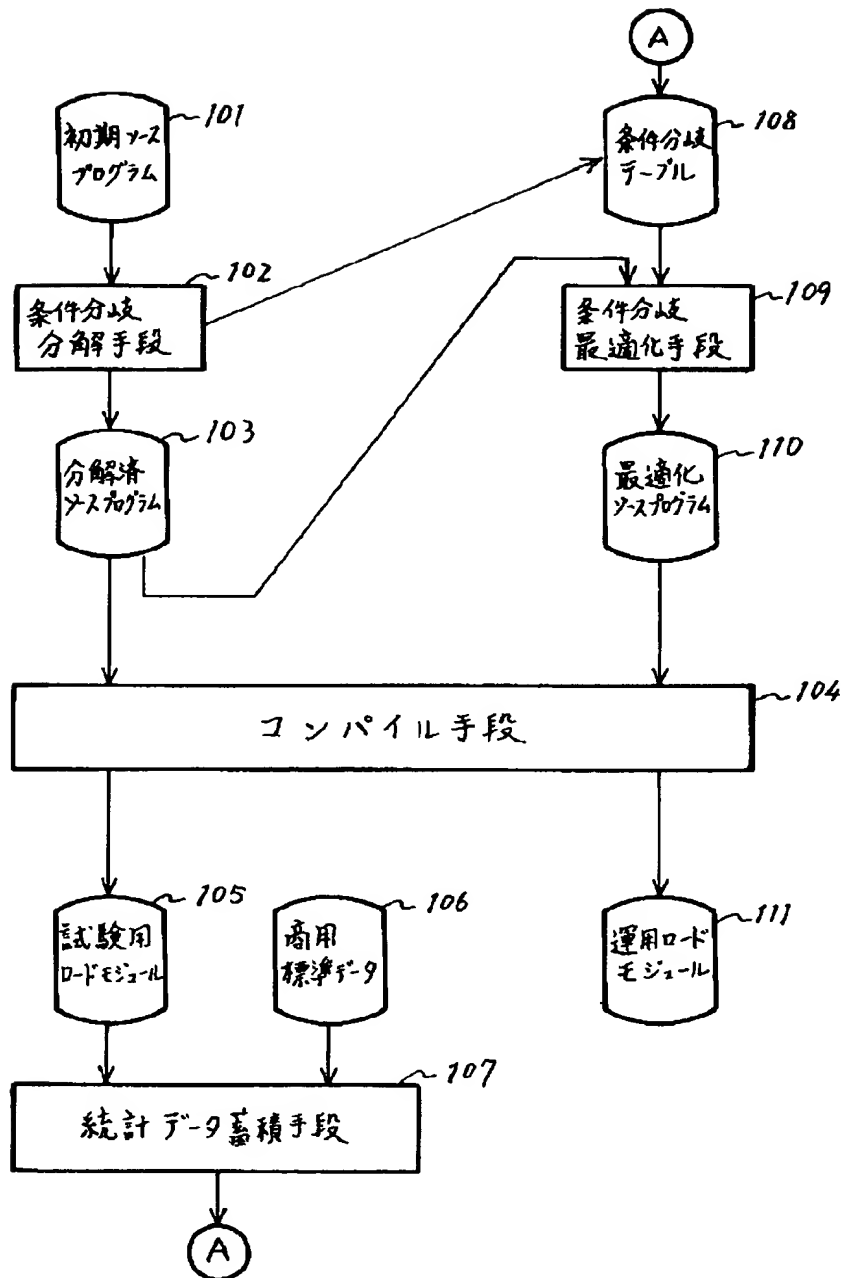
* 【図7】図7は本発明の一実施例における条件分岐最適化手段109の処理を示す流れ図である。

【符号の説明】

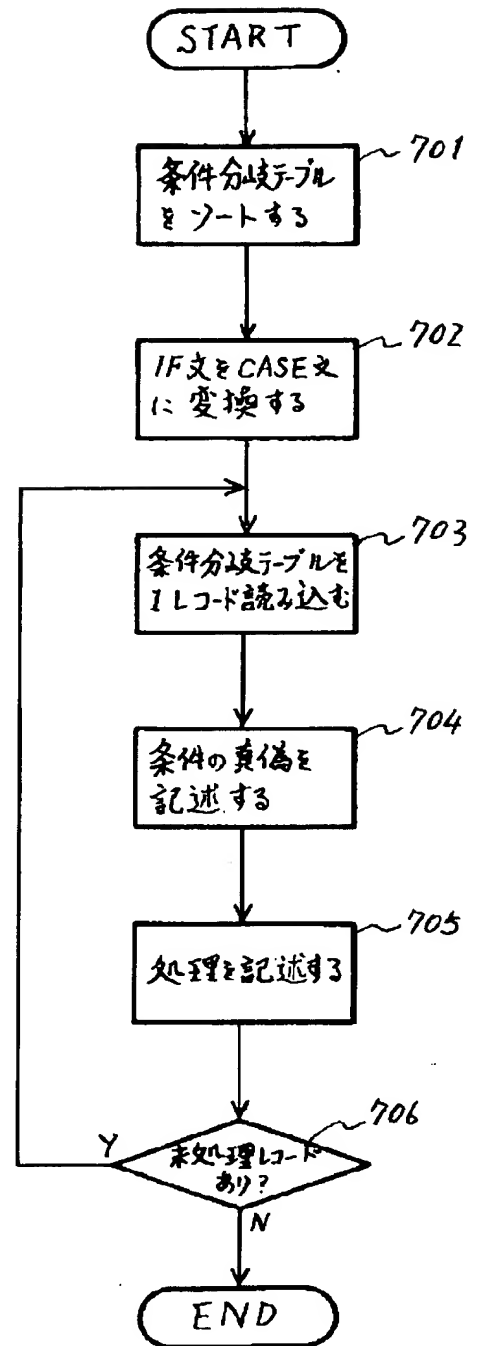
- 101 初期ソースプログラム
- 102 条件分岐分解手段
- 103 分解済ソースプログラム
- 104 コンパイル手段
- 105 試験用ロードモジュール
- 106 商用標準データ
- 107 統計データ蓄積手段
- 108 条件分岐テーブル
- 109 条件分岐最適化手段
- 110 最適化ソースプログラム
- 111 運用ロードモジュール

*

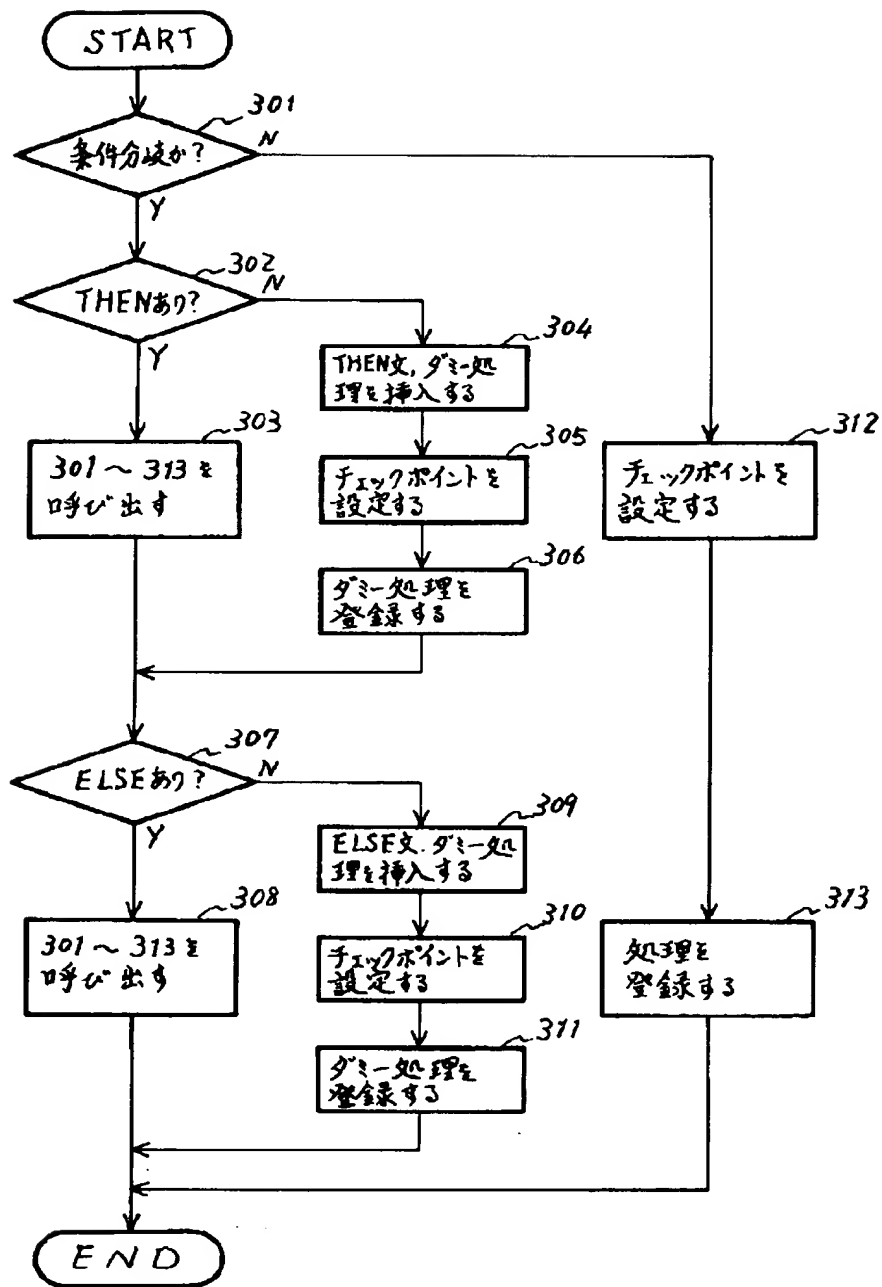
【図1】



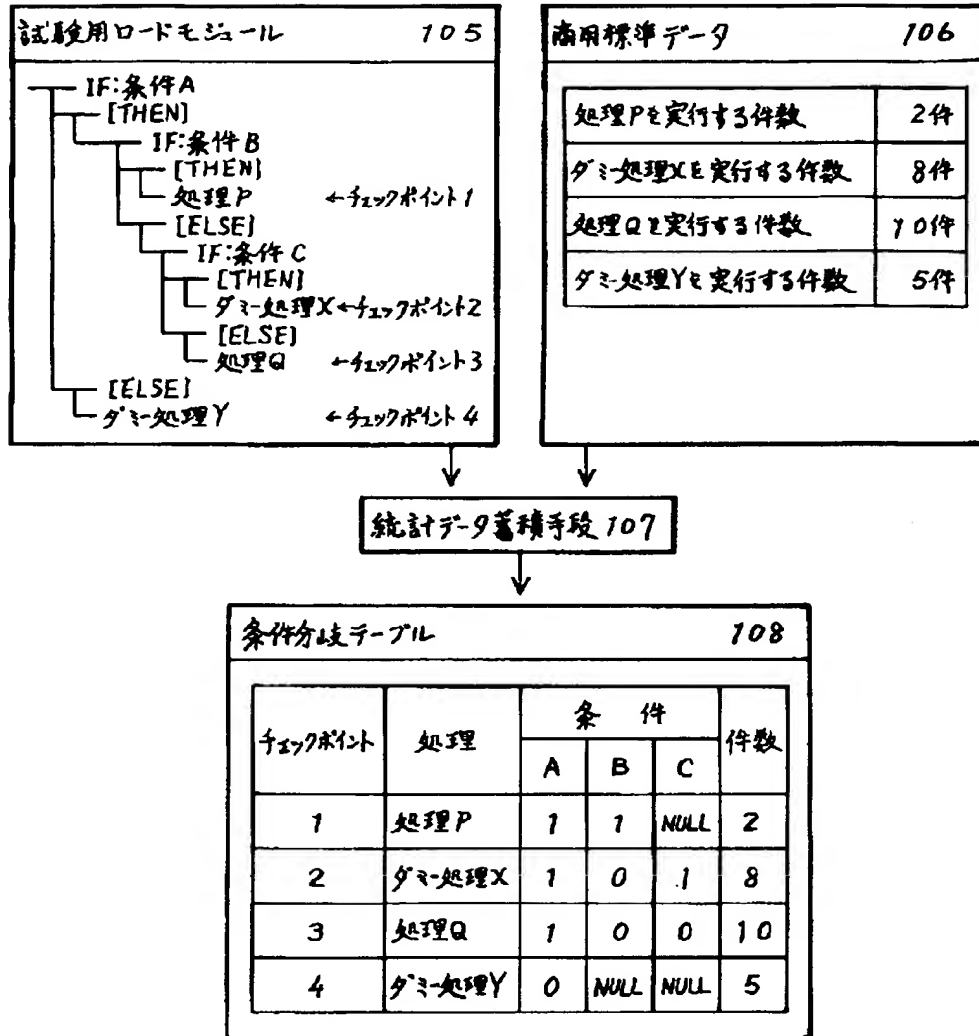
【図7】



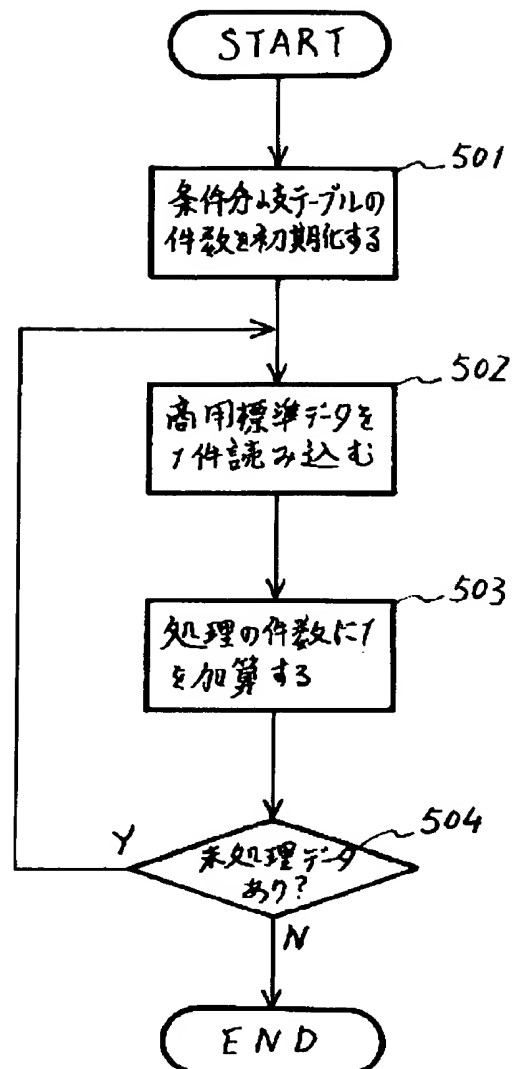
【図3】



【図4】



【図5】



【図6】

